REMARKS/ARGUMENTS

This Amendment and the following remarks are intended to fully respond to the Final Office Action dated July 27, 2005. In that Office Action, claims 1-37 were examined, and all claims were rejected. More specifically, claims 1-37 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Hara et al. (USPN 6,199,111), hereinafter "Hara," and Ote et al. (USPN 6,199,180), hereinafter "Ote". Reconsideration of these rejections, as they might apply to the original and amended claims in view of these remarks, is respectfully requested.

In this Response, claim 35 has been amended; no new claims have been added; and claim 18 and claim 33 have been canceled. Therefore, claims 1-17, 19-32, and 34-37 remain present for examination.

Claim Rejections – 35 U.S.C. § 103

Claims 1-37 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Hara and Ote. All amendments to the claims were made to place the claims in a more appropriate format and not to respond to any rejection by the Examiner. As such, the Applicant respectfully traverses the rejection since the Examiner has failed to establish a *prima facie* case of obviousness. In order to establish a *prima facie* case of obviousness, the Examiner must establish: 1) some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the references or combine their teachings; 2) a reasonable expectation of success of such a modification or combination; and 3) a teaching or suggestion in the cited prior art of each claimed limitation. See, MPEP §706.02(j). As will be discussed in detail below, the references cited by the Examiner fail to teach or suggest each claimed limitation. Specifically, the references, alone or in combination, fail to teach or suggest a machine automation client object class adapted to execute on the client machine in communication with an instance of the machine automation server object class, and the predefined machine automation client object class comprises one or more machine automation client objects for executing testing methods on the client machine or a machine automation control module instantiating a machine automation server object of the machine automation server object class in the server process and instructing the machine automation server object to instantiate one of the machine automation client objects of the machine automation client object class on the client machine to control the testing of the client machine.

Generally, the pending claims relate to automating control over one or more client machines, such as personal computers, using a server object of a predefined server object class, a client object of a predefined client object class, and a control module. The control module instantiates the server object in a server process, and the server object instantiates the client object on the client machine. As defined on page 8, lines 8 and 9 of the detailed description and consistent with common usage, objects are software modules that can encapsulate both data and functionality. Also as defined in the detailed description, initiation of an object can be accomplished by instantiating the object. See, e.g., pages 8-10 and 12-13. Instantiation includes creating and initializing the software object.

Hara "relates to a network computing system for updating and adding data in a distributed client-server system through a network." Col. 1, lines 6-8. Under Hara, a client system having a common communication unit communicates via a connection management unit with a plurality of servers each having a common communication unit. See, col. 4, lines 29-65. That is, Hara teaches data communications between a client and one or more servers through a connection management unit. However, as Examiner admits, Hara does not teach or suggest a machine automation client object class adapted to execute on the client machine in communication with an instance of the machine automation server object class, and the predefined machine automation client object class comprises one or more machine automation client objects for executing testing methods on the client machine or a machine automation control module instantiating a machine automation server object of the machine automation server object class in the server process and instructing the machine automation server object to instantiate one of the machine automation client objects of the machine automation client object class on the client machine to control the testing of the client machine. See, Office Action, Section 2.

Ote relates to "a manager for monitoring and controlling faults and performance of a plurality of computers on a network." Col. 1, lines 8-15. Under Ote, a service processor board (SVP) in the computer to be managed monitors faults in and controls power to the computer. A remote management computer communicates with the SVP via an asynchronous communications interface. Through the connection to the SVP, the remote computer can monitor faults in and control power to the computer to be managed. See generally, col. 3. Therefore, Ote teaches monitoring of and control of power to a client from a remote management computer through a SVP in the computer to be managed. However, contrary to the assertions of the Examiner, Ote does not teach or suggest a machine automation client object class adapted to execute on the client machine in communication with an instance of the machine automation server object class, and the predefined machine automation client object class comprises one or more machine automation client objects for executing testing methods on the client machine or a machine automation control module instantiating a machine automation server object of the machine automation server object class in the server process and instructing the machine automation server object to instantiate one of the machine automation client objects of the machine automation client object class on the client machine to control the testing of the client machine.

Claim 1, 16, 34, and 35 are directed to machine automation systems, methods, and computer readable medium for automating control of a client machine under control of a server process. Each claim, in general, recites a machine automation server object class adapted to execute in the server process, a predefined machine automation client object class adapted to execute on the client machine in communication with an instance of the machine automation server object, and a machine automation control module instantiating a machine automation server object of the machine automation server object class in the server process and instructing the machine automation server object to instantiate a machine automation client object of the machine automation client object class on the client machine to control operation of the client machine. Examiner admits that Hara does not teach these elements and relies on Ote to provide the required disclosure. Unfortunately, Ote falls short because Ote only teaches monitoring of and control of power to a client from a remote management computer through a SVP.

It is unclear what portions of the system described in Ote that the Examiner believes is the server process and which portions are the client process. Regardless, Ote does not teach all of the elements of the claims in any configuration that maintains a functioning system. Notably, Ote does not teach a machine automation control module that instantiates a machine automation server object, where both the machine automation control module and the machine automation server object operate on a server. First, Ote teaches a remote management computer (see, Figs. 1A and 1B, numerals 23 and 27) having a manager (see, Figs. 1A and 1B, numerals 241 and/or 242). See, col. 5, lines 9-14. Examiner seems to conclude that the managers are similar to the control module recited in the claims of the present invention. If this conclusion is made, then

Ote definitely never teaches the control module instantiating a server object. While Ote describes a fault manager and a power manager (see, Fig. 1B, numerals 2421 and 2422), Ote does not teach that the manager instantiates the fault manager or the power manager. See, col. 5, lines 34-41. In fact, it is unclear whether the manager, the fault manager, and the power manager are hardware or software functions. To presume that Ote teaches a control module that instantiates a server object is to employ impermissible hindsight to presume teachings that Ote simply does not provide.

Further, Ote teaches that the manager connects to an agent on the managed computer. See, col. 5, lines 9-14 and col. 5, lines 35-41. However, Ote does not teach that the manager, the fault manager, or the power manager instantiates the agent. Thus, without such teaching, Ote simply does not teach the server object instantiating an object of the client automation class as defined in the claims of the present invention. In fact, the agent appears to be a part of the software that is disconnected during a power fault and which is never re-instantiated by the manager, the fault manager, or the power manager after the fault. See, col. 5, lines 46-55.

Examiner also seems to interpret that the SVP manager is the control module. See, Office Action, Section 2, page 3. Again, while Ote describes a critical fault manager, a line manager, and a power manager (see, Fig. 1B, numerals 293, 291, and 292), Ote does not teach that the SVP manager instantiates the critical fault manager, the line manager, or the power manager. See, col. 5, lines 56-67. Again, it is unclear whether the SVP manager, the critical fault manager, the line manager, and the power manager are hardware or software functions. Thus, Ote does not teach a control module that instantiates a server object and to make such a conclusion again employs impermissible hindsight to presume teachings that Ote simply does not provide.

Ote does teach a power controller that is connected to the SVP manager. See, col. 5, lines 63-66. However, a server object, such as the SVP manager, never initiates the power controller. See, col. 5, line 62 – col. 6, line 5. While Ote is replete with instances where the power controller sends commands to and receives commands from various hardware systems (see, col. 5, lines 62 – col. 6, line 44; col. 8, lines 19-21), there is not a single mention, in Ote, of the power controller being instantiated by something similar to the machine automation server object. The SVP manager may communicate with the power controller, but the SVP manager does not initiate the power controller. In fact, the power controller is always powered (see, col. 4, lines 59-60), and thus, would never need to be instantiated or re-instantiated by a server object as claimed in the present invention. Thus, Ote actually teaches away from the present invention. Rather than create a software hierarchy that can overcome a reboot that deletes the client object, Ote teaches a system where the client object is always present to maintain communications with the server object, even during a power fault. See, col. 5, lines 46-55.

In summary, the combination of Hara and Ote are no more relevant to the independent claims than either reference alone since the references, alone or in combination, fail to teach or suggest all claimed limitations. Specifically, the references fail to teach or suggest a machine automation client object class adapted to execute on the client machine in communication with an instance of the machine automation server object class, and the predefined machine automation client object class comprises one or more machine automation client objects for executing testing methods on the client machine or a machine automation control module instantiating a machine automation server object of the machine automation server object class in the server process and instructing the machine automation server object to instantiate one of the machine automation client objects of the machine automation client object class on the client machine to control the testing of the client machine. As all other claims, i.e., claims 2-15, 17, 19-32, and 36-37, depend from the allowable independent claims, all claims are now in a condition for allowance and such action is respectfully requested.

Conclusion

This Amendment fully responds to the Office Action mailed on July 27, 2005. Still, that Office Action may contain arguments and rejections and that are not directly addressed by this Amendment due to the fact that they are rendered moot in light of the preceding arguments in favor of patentability. Hence, failure of this Amendment to directly address an argument raised in the Office Action should not be taken as an indication that the Applicants believe the argument has merit. Furthermore, the claims of the present application may include other elements, not discussed in this Amendment, which are not shown, taught, or otherwise suggested by the art of record. Accordingly, the preceding arguments in favor of patentability are advanced without prejudice to other bases of patentability.

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above remarks and amendments it is believed that the application is now in condition for allowance. Applicants request the application be allowed and pass to issuance as soon as possible. Should any additional issues need to be resolved, the Examiner is requested to telephone the undersigned attorney to resolve those issues.

Respectfully submitted,

Dated: 9/7/05

27488
PATENT TRADEMARK OFFICE

Tadd F. Wilson, Reg. No. 54,544
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903
303.357.1651